

## HOOFDSTUK 5: TAFELS OEFENEN

### TIJDSINDICATIE

- 70 minuten
- 80 minuten mét afspelen filmpje *Sandwichrobot* (let op: misschien heb je dat in Hoofdstuk 1 al gedaan)


### VOORKENNIS

- Onderwerpen uit Hoofdstukken 1 t/m 4
- Opslaan, rugzak, initiëren

### DOELEN TEN BEHOEVE VAN DE LEERLINGEN

- Tafelsommen maken. Daarbij gaat het de schrijvers van het boek om het proces. Maar misschien gaat het jouw klas/jouw school wel om het oefenen met tafels. Bij dit spel is sprake van integratie: programmeren wordt gecombineerd met een nuttige rekentool.
- Functie 'samenvoegen'
- Random: willekeurig. De computer willekeurige getallen laten kiezen. Toepassing en nut worden verderop toegelicht.
- Werken met geluiden, geluid opnemen en deze in je programma gebruiken.
- Variabelen

### LEERLIJN

- Abstractie, in de meest ruime zin. Kennisnet zegt daar o.a. over: *Leerling kan op een zodanige manier problemen formuleren dat het mogelijk wordt om het probleem op te lossen door gebruik van een computer of ander gereedschap.*
- Functie. Dit is een van de programmeertools uit het Blokkenpalet. Wij gebruiken de functie  om te checken of het door de gebruiker ingegeven antwoord overeenkomt met het juiste antwoord van de keer-som: in het linker deel komt de som te staan, het rechter deel bevat het door de gebruiker gegeven antwoord.

### LESVERLOOP, LES SUGGESTIES

- Leerlingen kunnen in tweetallen aan de slag. Paragraaf 'Wie ben je?' is een opwarmertje. Het samenvoegen van teksten wordt geoefend, zeg maar: het aan elkaar plakken van stukken tekst en variabelen. Het lastigste binnen deze opdracht is het opstellen van het algoritme waarbij de computerkeersommen genereert om te oefenen met tafels. Je zou leerlingen dit eerst met pen en papier kunnen laten doen: unplugged. Wellicht verstandig de les hiermee te beginnen! Dan weet je zeker dat er eerst is nagedacht, alvorens geprobeerd te hebben!
- De makkelijke weg is: het hoofdstuk stap voor stap doorwerken, daarmee 'kopiëren' de leerlingen het algoritme. Ze het algoritme zelf laten bedenken is écht lastig (maar wel mooi).
- Wil je dat laatste wél laten doen: eerst denken (pen en papier), dan doen (programmeren met Scratch). De sandwich robot is een voorbeeld van unplugged (zie Hoofdstuk 1). Vraag de leerlingen in tweetallen hoe het tafelspel er stap voor stap uitziet. Laat ze het op papier uitwerken, verwoord in 'computertaal'. Dit is waar programmeren om draait: denken, dan doen, stap voor stap je programma uitwerken op papier. Werk het uit in programmeertaal.

Voorbeeld:

*Persoon 1 is de computer, persoon 2 is de speler. Speel het stap voor stap. Neem bijvoorbeeld de tafel van 3 en doe 4 sommen. Op deze manier blijft het concreet. Dat is wel zo praktisch. Het kan er zo*

uitzien: Persoon 1 zegt “welke tafel wil je oefenen?”. Persoon 2 zegt “3”. Persoon 2 zegt “hoeveel sommen wil je oefenen?”. Persoon 2 zegt “4”. Denk nu na hoe de computer een som maakt, bijvoorbeeld:  $7 \times 3$ . Weet je dat de 7 hier willekeurig gekozen is door de computer? Speler 2 geeft antwoord. Dat antwoord kan goed of fout zijn. En zo ga je verder. Houd steeds in gedachten dat je in Scratch-taal moet denken.

- Snelle leerlingen kun je nog het aantal goed en fout beantwoorde vragen laten tonen. Toon ook het percentage vragen dat goed is beantwoord. Op deze manier krijgt programmeren toegevoegde waarde ten behoeve van rekenen en is er sprake van vak integratie.

## EVALUATIE

- Als je het unplugged door leerlingen laat uitvoeren, sta dan bij een paar zaken stil. We gaan er even van uit dat de tafel van 3 geoefend gaat worden. Laten we zeggen dat we 4 sommen willen. Hoe lang moet de lus zijn? Antwoord: net zo lang als jij hebt geantwoord op de vraag “hoeveel keersommen wil

je oefenen?” . De variabele Aantal sommen moet dus 4 worden

De variabele Tafel moet 3 worden

The image shows two Scratch code snippets. The first snippet consists of a 'herhaal' (repeat) block with a dropdown menu set to 'Aantal Sommen' and a 'vraag' (ask) block with the text 'Hoeveel sommen wil jij oefenen?' and an 'en wacht' (and wait) block. Below this is a 'maak' (make) block for the variable 'Aantal Sommen' with an 'antwoord' (answer) block. The second snippet is similar but the dropdown menu is set to 'Tafel' and the 'vraag' block text is 'Welke tafel wil jij oefenen?'.

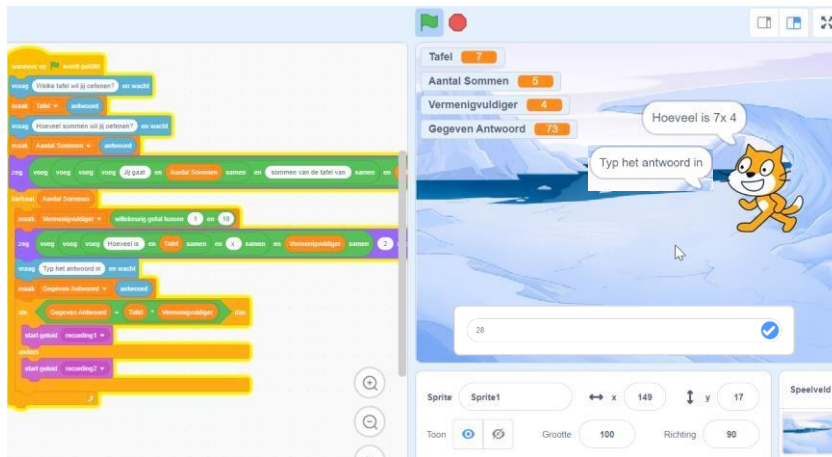
Wellicht is het lastig voor leerlingen te bedenken welke variabelen er moeten komen. Dat vergt gewoon veel oefening, voor nu: neem het over uit het boek. Net zoals Technic Lego: daarbij moet je ook eerst veel voorbeelden hebben nagemaakt alvorens je zelf iets kunt bouwen dat een functie heeft. Bij een vermenigvuldiging  $A \times B$ , heet  $A$  de *vermenigvuldiger* en  $B$  het *vermenigvuldigtal*. Bij de tafel van 3, krijg je  $A \times 3$ . De vermenigvuldiger  $A$  moet een willekeurig getal tussen 1 en 10 zijn. In het uitgewerkte voorbeeld is er een variabele gecreëerd met de naam Vermenigvuldiger

. Verder moet er een variabele Gegeven antwoord komen. Dat is het antwoord dat de gebruiker intypt bij de vermenigvuldig som

The image shows a Scratch 'maak' (make) block for the variable 'Vermenigvuldiger' with a 'willekeurig getal tussen' (random number between) block set to '1' and '10'.

The image shows a Scratch 'zeg' (say) block with the text 'Hoeveel is en Tafel samen en x samen en Vermenigvuldiger samen' and a '4 sec.' (4 seconds) block. Below this is a 'vraag' (ask) block with the text 'Typ het antwoord in' and an 'en wacht' (and wait) block. At the bottom is a 'maak' (make) block for the variable 'Gegeven Antwoord' with an 'antwoord' (answer) block.

- Let goed op dat het meegeven van waarden aan variabelen goed gaat, dat heb je vaak nodig bijprogrammeren.
- De vier variabelen die dus nodig zijn: Aantal sommen, Tafel, Vermenigvuldiger, Gegeven antwoord
- Stukken tekst aan elkaar koppelen met de samenvoeg-functie vergt nauwkeurig werken, zeker als ervaker dan één keer wordt samengevoegd.
- Als het opstellen met pen en papier van het tafelsom algoritme lastig blijkt: het boek werkt het stap voor stap uit. Dus mochten leerlingen er met pen en papier niet, of lastig, uitkomen: volg gewoon de uitwerkingen van het boek.
- Zo kán het resultaat van leerlingen er uit zien; het is slechts één van de vele voorbeelden.



## ACHTERGRONDINFORMATIE

Na het script  volgt altijd het  script.

Daarmee geef je de variabele de waarde mee die je net hebt ingetoetst. In dit geval: *naam* krijgt de waarde van wat jij hebt ingetoetst. Dit zul je vaak gebruiken!

De 'voeg samen' functie binnen onze zeg-boodschap  plakt stukken aan elkaar, in dit geval: het woord 'Hallo' en daar achter de waarde van de variabele. Dat betreft hier de naam die de gebruiker heeft ingevoerd. Je kunt zoveel stukken tekst en waarden van variabelen samenvoegen als je wilt.

## MEER INFORMATIE

- Abstractie is het zevende onderwerp beschreven in de leerlijn programmeren. Zie: [https://maken.wikiwijs.nl/74282/Programmeren\\_in\\_het\\_PO#!page-1838213](https://maken.wikiwijs.nl/74282/Programmeren_in_het_PO#!page-1838213)
- Functie is het achtste onderwerp beschreven in de leerlijn programmeren. Zie: [https://maken.wikiwijs.nl/74282/Programmeren\\_in\\_het\\_PO#!page-1838219](https://maken.wikiwijs.nl/74282/Programmeren_in_het_PO#!page-1838219)

Heb je ideeën, verbeteringen, aanvullingen of tips naar aanleiding voor Lesmethode programmeren met Scratch óf voor de lesbrieven? Mail naar: [info@POprogrammeren.nl](mailto:info@POprogrammeren.nl)